Action Anticipation By Predicting Future Dynamic Images

Cristian Rodriguez[1,2][0000−0002−2108−3904], Basura Fernando[1,2][0000−0002−6920−9916], and Hongdong Li[1,2][0000−0003−4125−1554]

[1] Australian National University, Canberra, Australia
[2] Australian Centre for Robotic Vision
cristian.rodriguez, basura.fernando, hongdong.li@anu.edu.au

Abstract. Human action-anticipation methods predict what is the future action by observing only a few portion of an action in progress. This is critical for applications where computers have to react to human actions as early as possible such as autonomous driving, human-robotic interaction, assistive robotics among others. In this paper, we present a method for human action anticipation by predicting the most plausible future human motion. We represent human motion using *Dynamic Images* [1] and make use of tailored loss functions to encourage a generative model to produce accurate future motion prediction. Our method outperforms the currently best performing action-anticipation methods by 4% on JHMDB-21, 5.2% on UT-Interaction and 5.1% on UCF 101-24 benchmarks.

Keywords: Action-Anticipation, Prediction, Generation, Motion Representation, Dynamic Image

1 Introduction

When interacting with other people, human beings have the ability to anticipate the behaviour of others and act accordingly. This ability comes naturally to us and we make use of it subconsciously. Almost all human interactions rely on this *action-anticipation* capability. For example, when we greet each other, we tend to anticipate what is the most likely response and act slightly proactively. When driving a car, an experienced driver can often predict the behaviour of other road users. Tennis players predict the trajectory of the ball by observing the movements of the opponent. The ability to anticipate the action of others is essential for our social life and even survival. It is critical to transfer this ability to computers so that we can build smarter robots in the future, with better social interaction abilities that think and act fast.

In computer vision, this topic is referred to as *action anticipation* [2–6] or early action prediction [7, 8]. Although action anticipation is somewhat similar to *action recognition*, they differ by the information being exploited. Action-recognition processes the entire action within a video and generate a category label, whereas action-anticipation aims to recognise the action *as early as possible*. More precisely, action-anticipation needs to predict the future action labels as early as possible by processing fewer image frames (from the incoming video), even if the human action is still in progress.

Instead of directly predicting action labels [4], we propose a new method that generates future motion representation from partial observations of human action in a video. We argue that the generation of future motion representation is more intuitive task than generating future appearance, hence easier to achieve. A method that is generating future appearance given the current appearance requires to learn a conditional distribution of factors such as colour, illumination, objects and object parts, therefore, harder to achieve. In contrast, a method that learns to predict future motion does not need to learn those factors. Furthermore, motion information is useful for recognising human actions [9, 10] and can be presented in various image forms [9, 11].

In this paper we propose to predict future motion representation for action anticipation. Our method hallucinates what is in the next motion representation of a video sequence given only a fraction of a video depicting a partial human action. We make use of a convolutional autoencoder network that receives a motion image as input at time t and outputs a motion image for the future (*e.g.* $t+1$). Using Markov assumption, we generate more motion images of the future using already generated motion images (*i.e.* we generate motion images for time $t+1, \cdots, t+k$). Then we process generated motion images using Convolutional Neural Network (CNN) to make action predictions for the future. As we are able to generate future motion images, now we are able to predict human actions only observing few frames of a video containing an action.

We train our action anticipation and motion generation network with several loss functions. These loss functions are specifically tailored to generate accurate representations of future motion and to make accurate action predictions.

Clearly, the motion information depends on the appearance and vice versa. For example, motion representations such as the optical flow relies on two consecutive RGB frames. Similarly, the content of dynamic images [9] relies on the appearance of consecutive frames. The relationship between static appearance and motion information is somewhat surprising and mysterious [12]. Recently, proposed dynamic images has managed to explore this relationship to some degree of success [9]. In particular, dynamic images summarise the temporal evolution of appearance of few frames (*e.g.* 10 frames) into a single image. Therefore, this motion summary image (a.k.a. dynamic image) captures the motion information of those frames. In this work, we hallucinate dynamic images for the future and use them for the task of action anticipation [3].

We generate dynamic images using both expected appearance and motion of the future. Specifically, future dynamic images are generated by taking into account both reconstructive loss (coined *dynamic loss*) and future expected appearance which is coined *static loss*. As motion and appearances should adhere

[3] However, the main concept of this paper is applicable for other types of motion images as well (optical flow, motion history images).

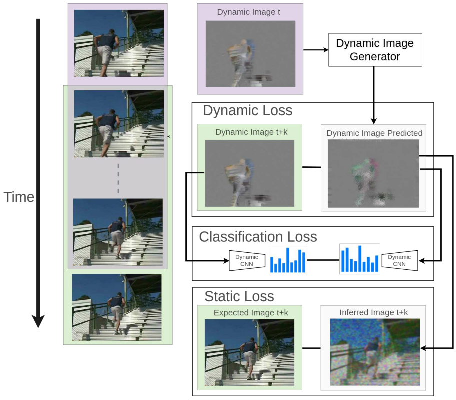


Fig. 1: Training of our generation module using multiple loss functions. **a)** ***Dynamic Loss*** evaluates the difference in motion information between predicted and ground truth dynamic image using $\mathcal{L}_2$ norm. **b)** ***Classification Loss*** takes care of generating dynamic images that are useful for action anticipation. **c) Static Loss** computes the $\mathcal{L}_2$ norm between predicted and ground truth RGB information at $t + k$ to evaluate the difference in appearance.

to each other, static loss is designed to satisfy expected future appearance in the generated dynamic images. In addition to that our generated dynamic images make use of class information and therefore discriminative. These loss functions are tailored to generate accurate future dynamic images as is depicted in Fig. 1. In a summary, we make the following contributions:

- Using a simple CNN architecture, we demonstrate the effectiveness of dynamic images for future content prediction.
- We design a set of effective loss functions to produce accurate future dynamic images.
- We obtain state-of-the-art performance for early activity recognition on standard benchmarks.

2 Related work

Action prediction and anticipation literature can be classified into deep learning and non-deep learning-based methods.

Human activity prediction is studied using integral histograms of spatial-temporal bag-of-features coined dynamic bag-of-words in the early days [3]. Yu *et al.* [13] propose to use spatial-temporal action matching for early action prediction task using spatial-temporal implicit shape models. Li *et al.* [14], propose to explore sequence mining where a series of actions and object co-occurrences are encoded as symbolic sequences. Kong *et al.* [15] explore the temporal evolution of human actions to predict the class label as early as possible. This model [15] captures the temporal dynamics of human actions by explicitly considering all the history of observed features as well as features in smaller temporal segments.

More recently, Soomro *et al.* [6] propose to use binary SVMs to localise and classify video snippets into sub-action categories and obtain the final class label in an online manner using dynamic programming. Because it is needed to train one classifier per sub-action, [5] extended this approach using a structural SVM formulation. Furthermore, this method introduces a new objective function to encourage the score of the correct action to increase as time progresses [5].

While all above methods utilise handcrafted features, most recent methods use deep learning approaches for action anticipation [2, 4, 16]. Deep learning-based methods can be primarily categorised into two types; 1. methods that rely on novel loss functions for action anticipation [2, 4, 17] and 2. methods that try to generate future content by content prediction [16].

In this context, [2] propose to use a Long Short-Term Memory (LSTM) with ranking loss to model the activity progression and use that for effective action prediction task. They use Convolutional Neural Network (CNN) features along with a LSTM to model both spatial and temporal information. Similarly, in [17], a new loss function known as the exponentially growing loss is proposed. It tries to penalize errors increasingly over time using a LSTM-based framework. Similarly, in [4], a novel loss function for action anticipation that aims to encourage correct predictions as early as possible is proposed. The method in [4] tries to overcome ambiguities in early stages of actions by preventing false negatives from the beginning of the sequence. Furthermore, a recently online action localisation method is presented which can also be used for online early action predictions [18]. However, this method primarily focuses on online action detection.

Instead of predicting the future class label, in [16], the authors propose to predict the future visual representation. However, the main motivation in [16] is to learn representations using unlabeled videos. Our work is different from [16] as we are predicting the future motion using dynamic images. We make use reconstruction loss, class information loss, and expected future appearance as a guide to predict future motion images. As our generated dynamic images are trained for action anticipation, they are class specific and different from original dynamic images [1]. As demonstrated, our generated dynamic images are more effective than original dynamic images for action anticipation task. Gao *et al.* [19] propose to generate future appearance using LSTM autoencoder to anticipate actions using both regression loss and classification loss. We argue that predicting future appearance representation is a complex task. We believe that action anticipation can benefit from motion prediction more than challenging appearance prediction.

Predicting the future content has been explored on other related problems in other domains of computer vision. Some of the work focuses on predicting (or forecasting) the future trajectories of pedestrians [20] or predicting motion from still images [20, 21]. However, we are the first to show the effectiveness of predicting good motion representations for early action anticipation.

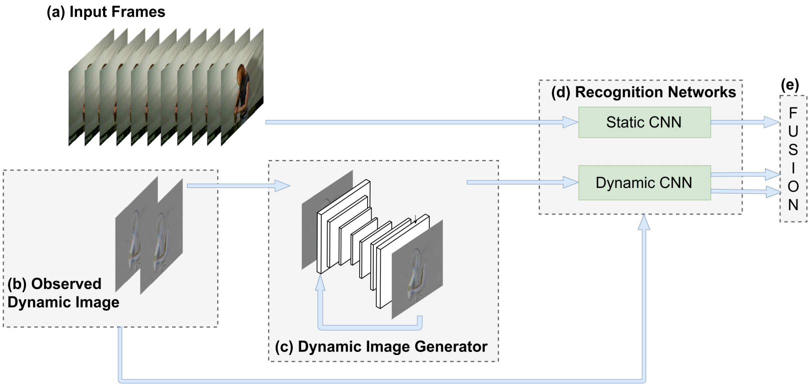


Fig. 2: **Overview of our approach**. We receive as an input a sequence of RGB video frames **(a)**. Then we use RGB images with windows size T to compute the Dynamic Images for seen part of the video **(b)**. The last dynamic image of the seen part is used to feed our dynamic image generator and generate $\hat{D}_{t+1}$ **(c)**. Next, we feed *Dynamic* CNN with observed dynamic images and generated dynamic images and *Static* CNN with RGB images **(d)**. Finally, we fusion all the outputs of our recognition networks **(e)**.

3 Method

The objective of our work is to recognise human actions as early as possible from a video sequence depicting human action. We present a method that hallucinates future motion from a partially observed human action sequence (RGB video clip). Then we process these hallucinated future motion representations to make future action predictions a.k.a. action anticipation. Our motion representation is based on dynamic images [1, 9]. Dynamic images model dynamic information of a short video clip and summarise motion information to a single frame. We present a method to hallucinate future dynamic images using a convolutional autoencoder neural network. We process generated dynamic images to predict future human actions using a CNN named *dynamic CNN*. To improve action recognition performance further, we use observed still image appearance information and process them with a *static CNN*. Furthermore, we make use of dynamic images created from observed RGB data and use the same dynamic CNN to make predictions. Therefore, we make use of three kinds of predictions and fuse them to make the final prediction (see Fig. 2). In the following section, we present some background about dynamic images 3.1 and then we present our dynamic image generation model in section 3.2. Then we discuss loss functions in section 3.3 and how to train our model in section 3.4.

3.1 Background

Dynamic images [1, 9] are a compact motion representation of videos which is useful for human action recognition. They summarise the temporal evolution of a short video clip (*e.g.* 10 frames) to a single still RGB image. Dynamic images are constructed using the rank pooling [22]. Rank pooling represents a video as a parameters of a linear ranking function that is able to chronologically order the elements of a sequence $\langle I_1, ..., I_T\rangle$. Precisely, let $\psi(I_t) \in \mathbb{R}^d$ be a feature vector extracted from each individual frame in the video and $V_t = \frac{1}{t}\sum_{\tau=1}^{t}\psi(I_\tau)$ be the average of these features up to time t. The ranking function $S(t|\mathbf{d})$ predicts a ranking score for each frame at time t denoted by $S(t|\mathbf{d}) = \langle \mathbf{d}, V_t\rangle$, where $\mathbf{d} \in \mathbb{R}^d$ is the parameter of the linear ranking function [22]. The parameter set $\mathbf{d}$ is learned so that the score reflect the rank of each frame. Therefore, the ranking score for later frame at time q ($q > t$) is associated with a larger score, *i.e.* $S(q|\mathbf{d}) > S(t|\mathbf{d})$. Learning $\mathbf{d}$ is posed as a convex optimisation problem using the RankSVM [23] formulation given as equation 1.

$$\begin{aligned} \mathbf{d}^* &= \rho(I_1, ..., I_t; \psi) = \underset{d}{\operatorname{argmin}}\, E(\mathbf{d}), \\ E(\mathbf{d}) &= \frac{\lambda}{2}||\mathbf{d}||^2 + \frac{2}{T(T-1)} \times \sum_{q>t} \max\{0, 1 - S(q|\mathbf{d}) + S(t|\mathbf{d})\}. \end{aligned} \tag{1}$$

Optimising equation 1 defines a function $\rho(I_1, ..., I_T; \psi)$ that maps a video sequence of length T to a single vector denoted by $\mathbf{d}$. Since this parameter vector contains enough information to rank all frames in the video clip, it aggregates temporal information from all frames. Therefore, it can be used as a video motion descriptor or a temporal descriptor.

When one applies this technique directly on RGB image pixels, the resulting $\mathbf{d}^*$ is known as the *dynamic image*. The output $\mathbf{d}^*$ has same dimensions as input images. Resulting dynamic image $\mathbf{d}^*$ summarises the temporal information of the RGB video sequence. Bilen *et al.* [1] present an approximation to rank pooling which is faster. This approximate rank pooling is essential for our method to hallucinate future dynamic images. Bilen *et al.* [1] proved that $\mathbf{d}^*$ can be expressed by the following equation 2.

$$\mathbf{d}^* = \sum_{t=1}^{T} \alpha_t I_t. \tag{2}$$

The coefficients α_t are given by $\alpha_t = 2(T - t + 1) - (T + 1)(H_T - H_{t-1})$ where $H_t = \sum_{i=1}^{t} 1/i$ is the t-th Harmonic number and $H_0 = 0$. We construct dynamic images using approximated rank pooling by taking a weighted sum of input image sequence where weights are given by predefined coefficients α.

3.2 Future motion prediction model

Given a collection of videos X with corresponding human action class labels Y, our aim is to predict the human action label as early as possible.

Each video $X_i \in X$ is a sequence of frames $X_i = \langle I_1, I_2, \cdots, I_n\rangle$ of variable length n. We process each sequence of RGB frames to obtain a sequence of dynamic images using equation 2. Instead of summarising the entire video with a single dynamic image, we propose to generate multiple dynamic images from a single video sequence using a fixed window size of length T. Therefore, each dynamic image is created using T consecutive frames. We process each training video X_i and obtain a sequence of dynamic images $\langle D_1, D_2, \cdots, D_n \rangle$. Our objective is to train a model that is able to predict the future dynamic image D_{t+k} given the current dynamic images up to time t *i.e.* $\langle D_1, D_2, \cdots, D_t \rangle$. Therefore, we aim to model the following conditional probability distribution using a parametric model

$$P(D_{t+k} | \langle D_1, D_2, \cdots, D_t \rangle ; \Theta) \tag{3}$$

where Θ are the parameters of our generative model ($k \geq 1$). We simplify this probabilistic model using the Markov assumption, hence now $k = 1$ and condition only on the previous dynamic image D_t. Then our model simplifies to following equation 4.

$$P(D_{t+1} | D_t; \Theta) \tag{4}$$

The model in equation 4 simplifies the training process. Furthermore, it may be possible to take advantage of different kinds of neural machine to implement the model in equation 4 such as autoencoders [24], variational conditional autoencoders [25, 26] and conditional generative adversarial networks [27].

Now the challenge is to find a good neural technique and loss function to train such a model. We use a denoising convolutional autoencoder to hallucinate future dynamic images given the current ones. Our convolutional autoencoder receives a dynamic image at time t and outputs a dynamic image for next time step $t+1$. In practice, dynamic images up to time t is observed, and we recursively generate dynamic images for time $t+1, \cdots, t+k$ using Markov assumption. Although we use a denoising convolutional autoencoder, our idea can also be implemented with other generative models. The autoencoder we use has 4 convolution stages. Each convolution has kernels of size 5×5 with a stride of 2 and the number of features maps for the convolution layers are set to 64, 128, 256, and 512 respectively. Then the deconvolution is the inverted mirror of the encoding network (see Fig 2), which is inspired by the architecture used in DCGAN [28]. Next, we discuss suitable loss functions for training the autoencoder.

3.3 Loss functions for training the autoencoder

First, we propose make use of reconstructive loss coined *Dynamic Loss* to reduce the $\mathcal{L}_2$ distance between predicted dynamic image $\hat{D}_{t+1}$ and the ground truth dynamic image obtained from the training data D_{t+1} as shown in equation 5.

$$\mathcal{L}_{DL} = ||\hat{D}_{t+1} - D_{t+1}||_2 \tag{5}$$

Even though this loss function helps us to generate expected future dynamic image, it does not guarantee that the generated dynamic image is discriminative for action anticipation. Indeed, we would like to generate a dynamic image that contains more action class information. Therefore, we propose to explore the teacher-student networks [29] to teach the autoencoder to produce dynamic images that would be useful for action anticipation. First, we train a teacher CNN which takes dynamic images as input and produces the action category label. Let us denote this teacher CNN by $f(D_i; \Theta_{cnn})$ where it takes dynamic image D_i and produces the corresponding class label vector $\hat{y_i}$. This teacher CNN that takes dynamic images as input and outputs labels is called *Dynamic CNN* (see Fig 2). This teacher CNN is trained with cross-entropy loss [30]. Let us denote our generator network as $g(D_t; \Theta) \rightarrow D_{t+1}$. We would like to take advantage of the teacher network $f(; \Theta_{cnn})$ to guide the student generator $g(D_t; \Theta)$ to produce future dynamic images that are useful for classification. Given a collection of current and future dynamic images with labels, we train the generator with the cross-entropy loss as follows:

$$\mathcal{L}_{CL} = -\sum_{t} y_i \log f(g(D_t; \Theta); \Theta_{cnn}) \tag{6}$$

where we fix the CNN parameter Θ_{cnn}. Obviously, we make the assumption that CNN $f(D_i; \Theta_{cnn})$ is well trained and has good generalisation capacity. We call this loss as the *classification loss* which is denoted by $\mathcal{L}_{CL}$. In theory, compared to original dynamic images [1, 9], our generated dynamic images are class specific and therefore discriminative.

Motion and appearance are related. Optical flow depends on the appearance of two consecutive frames. Dynamic images depends on the evolution of appearance of several consecutive frames. Therefore, it is important verify that generated future motion actually adhere to future expected appearance. Another advantage of using dynamic images to generate future motion is the ability exploit this property explicitly. We make use of future expected appearance to guide the generator network to produce accurate dynamic images. Let us explain what we mean by this. When we generate future dynamic image D_{t+1}, as demonstrated in equation 7, implicitly we also recover the future RGB frame I_{t+1}. Using this equation 7, we propose so-called *static loss* (SL) (equation 8) that consists of computing the $\mathcal{L}2$ loss between the generated RGB image $\hat{I}_{t+1}$ and real expected image I_{t+1}.

$$\begin{gathered} D_{t+1} = \sum_{i=1}^{T} \alpha_i I_{t+1+i} \\ D_{t+1} = \alpha_T I_{T+t+1} \sum_{i=1}^{T-1} \alpha_i I_{t+1+i} \\ I_{T+t+1} = \frac{D_{t+1} - \sum_{i=1}^{T-1} \alpha_i I_{t+1+i}}{\alpha_T} \end{gathered} \tag{7}$$

The applicability of static loss does not limit only to matching the future expected appearance, but also we guide the autoencoder model $g(; \Theta)$ to use all implicitly generated RGB frames from $\hat{I}_{t+2}$ to $\hat{I}_{T+t+1}$ making future dynamic image better by modeling the evolution of appearance of static images. Indeed, this is a better loss function than simply taking the dynamic loss as in equation 5.

$$\mathcal{L}_{SL} = ||\hat{I}_{T+t+1} - I_{T+t+1}||_2 \tag{8}$$

3.4 Multitask learning

We train our autoencoder with multiple losses, the static loss ($\mathcal{L}_{SL}$), the dynamic loss ($\mathcal{L}_{DL}$) and the classification loss ($\mathcal{L}_{CL}$). By doing so, we aim to generate dynamic images that are good for the classification, as well as representative of future motion. With the intention to enforce all these requirements, we propose to train our autoencoder with batch wise multitask manner. Overall, one might write down the global loss function $\mathcal{L} = \lambda_{sl}\mathcal{L}_{SL} + \lambda_{dl}\mathcal{L}_{DL} + \lambda_{cl}\mathcal{L}_{CL}$. However, instead of finding good scalar weights λ_{sl}, λ_{dl}, and λ_{cl}, we propose to divide each batch into three sub-batches, and optimise each loss using only one of those sub batches. Therefore, during each batch, we optimise all losses with different sets of data. We found this strategy leads to better generalisation than optimising a linear combination of losses.

3.5 Inference

During inference, we receive RGB frames from a video sequence as input. Using those RGB frames, we compute *dynamic images* following equation 2 with a window size length $T = 10$. In the case that the amount of frames is less that what is needed to compute the dynamic image i.e. 10% of the video is observed, we compute the dynamic image with the available frames according to equation 2. We use the last dynamic image (D_t) to predict the following dynamic image ($\hat{D}_{t+1}$). We repeat this process to generate k number of future dynamic images using Markov assumption. We process each observed RGB frame, observed dynamic images and generated dynamic images by respective static and dynamic CNNs that are trained to make predictions (see Fig. 2). Then, we obtain a score vector for each RGB frame, dynamic image and generated dynamic image. We sum them together and use temporal average pooling to make the final prediction.

4 Experiments and results

In this section, we perform a series of experiments to evaluate our action anticipation method. First, we present results for action recognition using the *static CNN* and the *dynamic CNN* in section 4.1. Then, we evaluate the impact of different loss functions for generating future dynamic images in section 4.2. After that in section 4.3, we compare our method with state-of-the-art techniques for action anticipation. Finally, we present some other additional experiments to further analyse our model in sections 4.4.

Datasets We test our method using three popular datasets for human action analysis JHMDB [31], UT-Interaction [32] and UCF101-24 [33], which have been used for action anticipation in recent prior works [4, 6, 18].

	JHMDB	**UT-Interaction**
Static CNN	55.0%	70.9%
Dynamic CNN	54.1%	71.8%

Table 1: Action recognition performance using dynamic and RGB images over JHMDB and UT-Interaction datasets. Action recognition performance is measured at frame level.

JHMDB dataset is a subset of the challenging HMDB51 dataset [34]. JHMDB is created by keeping action classes that involve a single person action. Videos have been collected from different sources such as movies and the world-wide-web. JHMDB dataset consists of 928 videos and 21 action classes. Each video contains one human action which usually starts at the beginning of the video. Following the recent literature for action anticipation [4], we report the average accuracy over the three splits and report results for so called *earliest* setup. For earliest recognition, action recognition performance is measured only after observing 20% of the video. To further understand our method, we also report recognition performance w.r.t. time (as a percentage). **UT-Interaction** dataset (UTI) contains 20 video sequences where the average length of a video is around 1 minute. These videos contain complete executions of 6 human interaction classes: shake-hands, point, hug, push, kick and punch. Each video contains at least one execution of an interaction, and up to a maximum of 8 interactions. There are more than 15 different participants with different clothing. The videos are recorded with 30fps and with a resolution of 720 x 480 which we resize to 320 x 240. To evaluate all methods, we use recommended 10-fold leave-one-out cross-validation per set and report the mean performance over all sets. **UCF101-24** dataset is a subset of the challenging UCF101 dataset. This subset of 24 classes contains spatio-temporal localisation annotation. It has been constructed for THUMOS-2013 challenge[4]. On average there are 1.5 action instances per video, each instance cover approximately 70% of the duration of video. We report the action-anticipation accuracy for set 1, as has been done previously in [18].

4.1 Training of *Static* and *Dynamic* CNNs.

In this section, we explain how we train our *static* and *dynamic* CNNs (see Fig. 2). Similar to [1, 9], we train a *Static CNN* for RGB frame-based video action recognition and a *Dynamic CNN* for dynamic image-based action recognition. In all our experiments, each dynamic image is constructed using 10 RGB frames (T=10). We use different data augmentation techniques to reduce the effect of over-fitting. Images are randomly flipped horizontally, rotated by a random amount in a range of -20 to 20 degrees, horizontally shifted in a range of -64 to 64 pixels, vertically shifted in a range of -48 to 48 pixels, sheared in a range of 10 degrees counter-clockwise, zoomed in a range of 0.8 to 1.2 and shifted channels in a range of 0.3. We make use of pre-trained Inception Resnet V2 [30] to fine-tune both *Static CNN* and the *Dynamic CNN* using a learning rate of

[4] http://crcv.ucf.edu/ICCV13-Action-Workshop/download.html

	JHMDB-21	UT-Interaction
$\mathcal{L}_{DL}$	42.8%	64.3%
$\mathcal{L}_{SL}$	49.5%	64.2%
$\mathcal{L}_{DL} + \mathcal{L}_{SL}$	53.4%	66.5%
$\mathcal{L}_{DL} + \mathcal{L}_{CL}$	52.5%	64.5%
$\mathcal{L}_{DL} + \mathcal{L}_{SL} + \mathcal{L}_{CL}$	54.0%	68.4%

Table 2: Results of using multitask learning to generate future dynamic images.

0.0001. We use a batch size of 32 and a weight decay of 0.00004. We use ADAM [35] optimizer to train these networks using epsilon of 0.1 and beta 0.5. Action recognition performance using these CNNs for JHMDB and UTI datasets are reported in Table 1. Note that the action recognition performance in Table 1 is only at frame level (not video level). We use these trained *Static* and *Dynamic* CNNs in the generation of future motion representation, dynamic images, and action anticipation tasks.

4.2 Impact of loss functions.

In this section we investigate the effectiveness of each loss function, explained in section 3.3, in the generation process of future dynamic images. We evaluate the quality of the generated dynamic images in a *quantitative* evaluation. Using the dynamic CNN to report action recognition performance over generated dynamic images.

We perform this experiment constructing a sequence of dynamic images using equation 2 for each test video in the dataset. Then for each test dynamic image, we generate the future dynamic image using our convolutional autoencoder. Therefore, the number of generated dynamic images is almost equal to real testing dynamic images. Then we use our dynamic CNN (which has been pretrained in previous section) to evaluate the action recognition performance of generated dynamic images (**DIg**). Using this approach we can evaluate the impact of several loss functions in the generation of dynamic images.

We use the first split of JHMDB and the first set of UTI to perform this experiment. We make use of the three proposed losses in section 3.3: dynamic-loss ($\mathcal{L}_{DL}$), class-based loss ($\mathcal{L}_{CL}$) and static-loss ($\mathcal{L}_{SL}$) to train our autoencoder. We train the convolutional autoencoder using ADAM solver with a batch size of 32, a learning rate of 0.0001. We train our model for 30 epochs using the same augmentation process used in section 4.1.

We use the generalisation performance of *real dynamic images* from Table 1 as a reference to estimate the quality of generated dynamic images. Since, we measure the performance of generated dynamic images in the same way.

As can be seen in Table 2, a combination of $\mathcal{L}_{DL}$, $\mathcal{L}_{CL}$ and $\mathcal{L}_{SL}$ gives excellent recognition performance of 54.0% for the generated dynamic images which is very close to the model performance of single dynamic CNN 54.1% in the case of JHMDB dataset. Indicating that our generative model along with loss functions are capable of generating representative and useful future dynamic images. A similar trend can be seen for UTI dataset. Notice that the $\mathcal{L}_{DL}$ and $\mathcal{L}_{SL}$ already

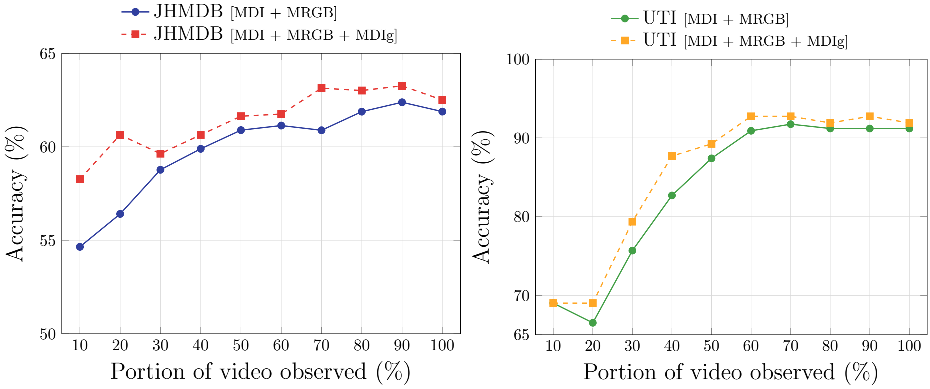


Fig. 3: Action anticipation performance with respect to portion of the video observed on JHMDB *(left)* and UTI *(right)* datasets.

produce good recognition performance on JHMDB and UTI datasets, which suggest that those losses can generated images that understand the human motion. However, those generated images are not class specific. We conclude that convolutional autoencoder model trained with three losses is able to generate robust future dynamic images. These generated dynamic images are effective in action recognition.

4.3 Action anticipation

Our action anticipation network consist of a *static* CNN and a *dynamic* CNN (see Fig 2). Our action anticipation baseline uses observed multiple RGB frames and multiple dynamic images similar to [1]. In addition to that our method generates K number of future dynamic images and make use of them with dynamic CNN. Action anticipation performance is evaluated at different time steps after observing fraction of the video (*i.e.*, 10%, 20% of the video). Results are shown in Figure 3, where we can see the effect of adding generated dynamic images (MDIg) to our pipeline. In the case of JHMDB the most significant improvement is obtained at 20% which is an enhancement of **5.1%** with respect to the baseline. In the UTI dataset, the most significant improvement is obtained at 40% of the video observed with a performance enhancement of **5.0%** with respect to the baseline. Moreover, the less significant improvement are obtained when the video observation approaches the 100% with a 0.62% and 0.71% of improvement with respect to the baseline on JHMDB and UTI dataset respectively.

Another standard practice is to report the action anticipation performance using *earliest* and *latest* prediction accuracies as done in [3, 4]. Although, there is no agreement of what is the proportion of frames used in earliest configuration through different datasets. We make use of the proportion that has been employed by baselines (20% and 50% of the video for JHMDB and UTI respectively). Therefore, following [4] we report results in Table 3, Table 5 for JHMDB and UTI datasets respectively. We outperform other methods that rely

Method	Earliest	Latest
DP-SVM [5]	5%	46%
S-SVM [5]	5%	43%
Where/What [6]	12%	43%
Context-Aware+Loss of [17]	28%	43%
Ranking Loss [2]	29%	43%
Context-Aware+Loss of [2]	33%	39%
E-LSTM [4]	55%	58%
ROAD [18]	57%	68%
Ours	**61%**	63%

Table 3: Comparison of action anticipation methods on **JHMDB** dataset. 20% of video is observed at *Earliest*.

	Earliest	Latest
ROAD (RTF) [18]	81.7%	83.9%
ROAD (AF) [18]	84.2%	85.5%
Ours	89.3%	90.2%

Table 4: Comparison of action anticipation methods on **UCF101-24** dataset. 10% of video is observed at Earliest.

Method	Earliest	Latest
S-SVN [5]	11.0%	13.4%
DP-SVM [5]	13.0%	14.6%
CuboidBayes [3]	25.0%	71.7%
CuboidSVM [36]	31.7%	85.0%
Context-Aware+Loss of [17]	45.0%	65.0%
Context-Aware+Loss of [2]	48.0%	60.0%
BP_SVM [37]	65.0%	83.3%
I-BoW [3]	65.0%	81.7%
D-BoW [3]	70.0%	85.0%
E-LSTM [4]	84.0%	90.0%
Ours	89.2%	91.9%

Table 5: Comparison of action anticipation methods using **UTI** dataset. 50% of the video is observed at *Earliest*.

on additional information, such as optical flow [2, 5, 6] and Fisher vector features based on improved Dense Trajectories [5]. Our approach outperforms the state-of-the-art by **4.0%** on JHMDB and by **5.2%** on UTI datasets in the earliest configuration. Finally, we report results on UCF101-24 dataset for action anticipation. For this dataset, we use 10% of the video to predict the action class in the earliest configuration. As we can see in Table 4, We outperform previous method [18] by **5.1%** on the earliest configuration. A more detailed analysis using UCF101-24 dataset is provided on the supplementary material.

These experiments evidence the benefits of generating future motion information using our framework for action anticipation.

4.4 Further exploration

In Fig. 4 we observe the influence of generating dynamic images recursively for earliest configuration in JHMDB and UTI datasets. We generate K number of future dynamic images recursively using the very last true dynamic image. As it can be seen in Fig. 4, as we generate more dynamic images into the future, the prediction performance degrades due to the error propagation. We report action recognition performance for each generated future dynamic image (*i.e.* for the generated future dynamic image at K). If we do not generate any dynamic image for the future, we obtain an action recognition performance of 55.9%. If we include generated dynamic images, we obtain a best of 61.0% on JHMDB. A similar trend can be seen for UTI dataset, where without future dynamic image we obtain 87.4% and after generation we obtain an action recognition performance of 89.2%. The influence of generating more future dynamic images is shown in Fig 4.

Fig. 4: Impact of generating more future dynamic images recursively on JHMDB *(left)* and UTI *(right)* datasets. K is the number of generated dynamic images based on observed RGB frames. K=0 means no dynamic image is generated.

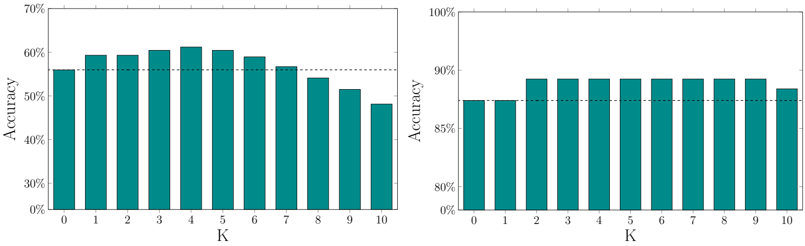
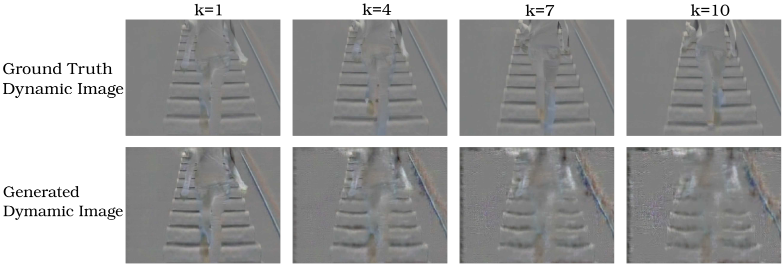


Fig. 5: Visual comparison between generated dynamic image *(bottom)* and ground truth *(top)*. K refers to how many iterations we apply in the generation of dynamic image.



Finally, we visually inspect the recursively generated dynamic images for K equal to 1, 4, 7 and 10 in Fig. 5. Although, we can use our model to generate quite accurate dynamic images, as we predict into the further, the generated dynamic images might contain some artifacts.

5 Discussion

In this paper, we demonstrate how to hallucinate future video motion representation for action anticipation. We propose several loss functions to train our generative model in a multitask scheme. Our experiments demonstrate the effectiveness of our loss functions to produce better future video representation for the task of action anticipation. Moreover, experiments show that made use of the hallucinated future video motion representations improves the action anticipation results of our powerful backbone network. With our simple approach we have outperformed the state-of-the-art in action anticipation in three important action anticipation benchmarks. In the future, we would like to incorporate additional sources of information to hallucinate other dynamics such as optical flow using the same framework. Furthermore, we would like to extend this method to predict dynamic images further into the future.

Acknowledgments

We thank NVIDIA Corporation for the donation of the GPUs used in this work.

References

1. Bilen, H., Fernando, B., Gavves, E., Vedaldi, A., Gould, S.: Dynamic image networks for action recognition. In: CVPR. (2016)
2. Ma, S., Sigal, L., Sclaroff, S.: Learning activity progression in lstms for activity detection and early detection. In: CVPR. (2016)
3. Ryoo, M.S.: Human activity prediction: Early recognition of ongoing activities from streaming videos. In: ICCV. (2011)
4. Sadegh Aliakbarian, M., Sadat Saleh, F., Salzmann, M., Fernando, B., Petersson, L., Andersson, L.: Encouraging lstms to anticipate actions very early. ICCV (2017)
5. Soomro, K., Idrees, H., Shah, M.: Online localization and prediction of actions and interactions. arXiv:1612.01194 (2016)
6. Soomro, K., Idrees, H., Shah, M.: Predicting the where and what of actors and actions through online action localization. In: CVPR. (2016)
7. Lan, T., Chen, T.C., Savarese, S.: A hierarchical representation for future action prediction. In: ECCV. (2014)
8. Yu, G., Yuan, J., Liu, Z.: Predicting human activities using spatio-temporal structure of interest points. In: ACMMM. (2012)
9. Bilen, H., Fernando, B., Gavves, E., Vedaldi, A.: Action recognition with dynamic image networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **PP**(99) (2017) 1–1
10. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NIPS. (2014)
11. Ahad, M.A.R., Tan, J.K., Kim, H., Ishikawa, S.: Motion history image: its variants and applications. Machine Vision and Applications **23**(2) (2012) 255–281
12. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR. (2017)
13. Yu, G., Yuan, J., Liu, Z.: Predicting human activities using spatio-temporal structure of interest points. In: ACMMM. (2012)
14. Li, K., Fu, Y.: Prediction of human activity by discovering temporal sequence patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(8) (2014) 1644–1657
15. Kong, Y., Kit, D., Fu, Y.: A discriminative model with multiple temporal scales for action prediction. In: ECCV. (2014)
16. Vondrick, C., Pirsiavash, H., Torralba, A.: Anticipating visual representations from unlabeled video. In: CVPR. (2016)
17. Jain, A., Singh, A., Koppula, H.S., Soh, S., Saxena, A.: Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In: ICRA. (2016)
18. Singh, G., Saha, S., Sapienza, M., Torr, P.H.S., Cuzzolin, F.: Online real-time multiple spatiotemporal action localisation and prediction. In: ICCV. (2017)
19. Gao, J., Yang, Z., Nevatia, R.: Red: Reinforced encoder-decoder networks for action anticipation. arXiv:1707.04818 (2017)
20. Kitani, K.M., Ziebart, B.D., Bagnell, J.A., Hebert, M.: Activity forecasting. In: ECCV. (2012)
21. Pellegrini, S., Ess, A., Schindler, K., Van Gool, L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: ICCV. (2009)
22. Fernando, B., Gavves, E., Oramas, J., Ghodrati, A., Tuytelaars, T.: Rank pooling for action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(4) (2017) 773–787

23. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Statistics and computing **14**(3) (2004) 199–222
24. Baldi, P.: Autoencoders, unsupervised learning, and deep architectures. In: ICML. (2012)
25. Kingma, D.P., Mohamed, S., Rezende, D.J., Welling, M.: Semi-supervised learning with deep generative models. In: NIPS. (2014)
26. Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. In: NIPS. (2015)
27. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv:1411.1784 (2014)
28. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. ICLR (2016)
29. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv:1503.02531 (2015)
30. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI. (2017)
31. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M.J.: Towards understanding action recognition. In: ICCV. (2013)
32. Ryoo, M.S., Aggarwal, J.K.: UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html (2010)
33. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv:1212.0402 (2012)
34. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: a large video database for human motion recognition. In: ICCV. (2011)
35. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. ICLR (2015)
36. Ryoo, M., Chen, C.C., Aggarwal, J., Roy-Chowdhury, A.: An overview of contest on semantic description of human activities (sdha) 2010. In: ICPR. (2010)
37. Laviers, K., Sukthankar, G., Aha, D.W., Molineaux, M., Darken, C., et al.: Improving offensive performance through opponent modeling. In: AIIDE. (2009)